Felix Klose
klose@cg.cs.tu-bs.de
Computer Graphics Lab, TU Braunschweig

Christian Linz
linz@cg.cs.tu-bs.de
Computer Graphics Lab, TU Braunschweig

Christian Lipski
lipski@cg.cs.tu-bs.de
Computer Graphics Lab, TU Braunschweig

Prof. Dr. Ing. Marcus Magnor
magnor@cg.cs.tu-bs.de
Computer Graphics Lab, TU Braunschweig

Flexible Stereoscopic 3D Content Creation of Real World Scenes

Technical Report 2010-11-14

November 26, 2010

Computer Graphics Lab, TU Braunschweig

Abstract

We propose an alternative over current approaches to stereoscopic 3D video content creation based on a free-viewpoint video. Acquisition and editing is greatly simplified. Our method is suitable for arbitrary real-world scenes. From unsynchronized multi-view video footage, our approach renders high-quality stereo sequences without the need to explicitly reconstruct any scene depth or geometry. By allowing to freely edit viewpoint, slow motion, freeze-rotate shots, depth-of-field, and many more effects, the presented approach extends the possibilities in stereo 3D movie creation.

1 Introduction

The growing popularity of stereoscopic 3D in movies and television is creating a demand for easy content creation. Three major methodologies for creating stereoscopic material are currently employed: purely digital content creation for animation films and games, content filmed with specialized stereoscopic cameras, and stereo hallucination from monocular video.

If the entire production process is digital in the sense that the shown scene exists as computer graphics models, the creation of a stereoscopic image pair is straightforward. Instead of rendering one image per video frame, a second image with a shifted virtual viewpoint is rendered. Since the recording environment including time is fully controllable, dynamic scenes do not pose an additional problem. The major drawback is, that the creation of naturalistic real world images is extremely complex and time consuming.

The enhancement of monocular recordings suffers from a similar problem. Although the recorded footage has the desired natural look, the creation of a proxy geometry or a scene model can be tedious. The depth map or proxy geometry used to synthesize a second viewpoint has to be created by hand modeling. Therefore the complexity of the scene model creation directly depends on the complexity of the recorded scene.

While directly recording with a stereoscopic camera eliminates the need to create an additional scene model, it requires the highly specialized and therefore expensive stereo-camera hardware. Leaving aside monetary constraints, the on set handling of the stereoscopic cameras poses a challenge. The view and baseline selection for example requires careful planning to give the viewer a pleasing stereoscopic experience. Changing the parameters in post production again is difficult or even impossible.

We propose to approach the problem from a purely image-based rendering perspective and thereby eliminate the requirement for an explicit scene model. By using multi-view datasets of loosely aligned cameras, it becomes possible to interpolate arbitrary stereoscopic views between the input camera viewpoints. If the camera position is controllable after the data acqui-

sition, view and baseline selection can happen in the post processing stage. The term free-viewpoint video has been coined for this method in the monocular case, which we extend to the stereoscopic case. Since our image-based stereoscopic free-viewpoint video framework is capable of time and space interpolation, it combines the natural image impression from direct stereoscopic recording with the full viewpoint and time control of digitally created scenes. Further on, having full control over time and camera viewpoint, effect shots as a freeze-rotate are easily possible without the need for complex hardware setups. Even more stereoscopic special effects in the spatial or temporal domain are possible within this purely image-based framework.

In Sect. 2 we give an overview of the methods and current research relevant for this work. We then discuss the preprocessing of the input data in Sect. 3. The image formation and image-based rendering algorithm is described in Sect. 4. In Sect. 5 we show results that were created using only image-based techniques. For different scenarios geometrically plausible stereoscopic renderings are shown. We conclude in Sect. 6.

2 Related Work

Stereoscopy. Since its invention in 1838, stereoscopy has been widely used in photography and film making industry. It has recently received renewed attention, partly because sophisticated stereoscopic equipment became available for the consumer market. Although the basic principle of stereoscopic image acquisition seems quite simple, many pitfalls exist that make stereoscopic capture a tedious task. A good introduction to stereoscopic movie making is given by Mendiburu [23]. This book formalizes the concepts with clear and simple drawings and is considered as the reference in the movie-making community. Similarly, Devernay and Beardsley give an in-depth review on the state of understanding in stereoscopic cinema Devernay and Beardsley [5]. In their article, they discuss perceptual factors, choice of camera geometry at capture time and post-production tools to manipulate 3D experience. A good introduction to current stereoscopic editing techniques can be found in Wilkes [30]. Typical stereoscopic editing task are image rectification, color balancing, disparity remapping and baseline editing. The latter one is especially interesting for our approach, since multi-view recordings often feature wide baselines and conversion to stereoscopic output material is not straight-forward. Rogmans et al. reviewed the available methods for novel view synthesis from stereoscopic data, and noticed that they essentially consist of two steps: disparity estimation and view synthesis Rogmans et al. [24]. Since disparity estimation is often error-prone, Devernay and Peon proposed a novel view synthesis for altering interocular distance with on-the-fly artifact detection and removal Devernay and Peon [7]. Disparity remapping also recently received

considerable attention: Lang et al. proposed non-linear disparity mapping operators to alter perceived scene depth, necessary for content adaption to different viewing geometries Lang et al. [15]. Targeting the same application, Devernay and Duchêne proposed a disparity-remapping scheme that does not distort objects with respect to perceived depth Devernay and Duchêne [6]. To aid the stereographer in the first place with capture, Zilly et al. presented the Stereoscopic analyzer, a tool for online validation of stereoscopic capture, including image rectification, detection of window violations, and optimal interocular distance proposal Zilly et al. [32]. Most similar to our proposed approach to stereoscopic content creation is the work of Guillemaut et al. [13]. They reconstruct a high-quality scene geometry from wide-baseline multi-view footage and use this geometry for stereoscopic view synthesis. While giving good results, they only demonstrated their approach on indoor scenes.

Image Morphing. Image morphing is a technique that accepts at least two input images and lets the user create a visually plausible in-between representation. Image morphing requires pixel correspondences between input images. Traditionally, these are created in a user-assisted workflow Wolberg [31] or are derived from other data, such as depth maps Chen and Williams [3], Didyk et al. [8]. Seitz and Dyer Seitz and Dyer [25] extended the original forward-warping and blending technique to produce geometrically plausible results. We employ their proposed image reprojection to align our input data and to produce the desired output, i.e., parallel or converging stereoscopic views. Lee at al. Lee et al. [16] described how to make use of more than two input images. In order to cope with potentially unsynchronized video data, we pick up on their idea and interpolate both view direction and recording time.

Optical Flow. Optical flow algorithms are able to compute dense pixel correspondences between two input images. These correspondences may serve as input for image morphing and make user-driven correspondence estimation redundant. A survey on recent optical flow algorithm was composed by Baker et al. Baker et al. [1]. Recently, optical flow approaches have been proposed which are specifically tailored for the task of image morphing. Stich et al. Stich et al. [28; 27] designed a perceptually inspired optical flow algorithm for view interpolation. Lipski et al. Lipski et al. [20] introduced representative SIFT descriptors for high-resolution image correspondence estimation. Linz et al. Linz et al. [17] combined the two latter approaches with a gradient-domain based rendering Mahajan et al. [21].

Free-Viewpoint Video Systems. Free-viewpoint video systems render novel views from multi-video data. Although many approaches with con-

vincing results have been proposed so far, rendering of stereoscopic views remains an open problem.

The first category of free-viewpoint video systems relies on a geometric reconstruction of the scene. Although stereoscopic rendering is straightforward if the scene geometry is known, they all suffer from typical drawbacks of geometry-based systems. Zitnick et al. Zitnick et al. [33] achieve view interpolation from synchronously captured multi-view video data. Unfortunately, time interpolation is not possible with their approach and cameras have to be densely spaced. De Aguiar et al. de Aguiar et al. [4] present a high-quality performance-capturing that requires the exact knowledge of the 3D geometry of the performing actor. Eisemann et al. Eisemann et al. [10] show that misaligned reprojected textures can be corrected on-the-fly with image-based techniques. However, they assume that the overall object silhouette is faithfully preserved. Guillemaut et al. allow free-viewpoint rendering of sports events, e.g. soccer or rugby games Guillemaut et al. [12]. They use a segmentation and reconstruction approach, relying on the known color distribution of the background. Scene flow estimation enables the reconstruction of object movement Vedula et al. [29]. Klose et al. Klose et al. [14] designed a scene flow reconstruction that is able to cope with unsynchronized multi-view recordings. However, their estimation produces only quasi-dense information and does not recover a valid model in textureless regions.

Recently, image-based approaches have been introduced to the community that circumvent the problems of geometry reconstruction. Germann et al. Germann et al. [11] represent soccer players as a collection of articulated billboards. Their free-viewpoint system is restricted to scenes with a well-defined background, e.g., soccer stadiums. Ballan et al. Ballan et al. [2] present an image-based view interpolation that uses billboards to represent a moving actor. Although they produce good results for a single rendered viewport, their billboard-based technique is not suitable for stereoscopic rendering. Lipski et al. propose an image-based free-viewpoint system Lipski et al. [19] that is based upon multi-image-morphing. They accept unsynchronized multi-view recordings as their input and interpolate both viewpoints as well as time. Their approach has also been extended to produce a variety of visual effects Linz et al. [18]. We base upon their work and show that not only visually plausible, but also geometrically valid results can be created using a purely image-based approach.

3 Data Preprocessing

The algorithm we use processes multi-view sequences recorded by consumer grade video cameras. The cameras are set up with relatively wide baselines and directed at a convergence point. Only a rough alignment and no on

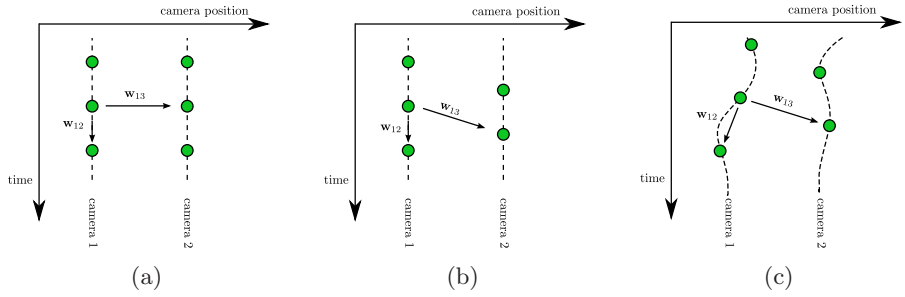


Figure 1: The layout of images shown as green dots in the space-time plane with different camera configurations: (a) static and synchronized cameras (b) static unsynchronized cameras (c) moving unsynchronized cameras.

set calibration of the cameras is performed. The goal is solely to ensure a sufficient image overlap between adjacent cameras. In all our test sequences the camera shutters are unsynchronized, therefore no specialized hardware is needed.

In the following section we describe the embedding of the input images in a low dimension navigation space to allow intuitive scene navigation.

3.1 Camera Alignment

An offline processing estimates the extrinsic camera parameters $\mathbf{R}$ and $\mathbf{p}$ including acquisition time t. The extrinsic camera parameters in conjunction with the image acquisition time form a high dimensional space. However, the goal of free-viewpoint navigation is to explore a scene captured by multiple video cameras in an intuitive way. Practical parameterizations that allow for intuitive navigation and realistic view interpolation require the cameras' optical axes to cross at a convergence point (possibly at infinity). A natural choice for an embedding into a parameter space is a spherical parameterization of the camera setup. A spherical model is sufficiently expressive to account for a large variety of physical camera setups, ranging from cameras arranged in a one-dimensional arc, over cameras placed in a spherical setup to linear camera arrays with parallel optical axes.

To this end we choose to define a three-dimensional navigation space $\mathcal{N}$ that represents spatial camera coordinates as well as the temporal dimension. Cameras are placed on the surface of a virtual sphere, their orientations are defined by azimuth φ and elevation θ. Together with the temporal dimension t, φ and θ span a three-dimensional navigation space $\mathcal{N}$. A novel (virtual) view of the scene $I_v(\varphi, \theta, t)$ is a function of position in navigation space. If cameras are arranged in an arc around the scene, θ is fixed, reducing $\mathcal{N}$ to two dimensions.

To serve as sampling points, the camera configuration of the recorded multi-video input in Euclidean world space is embedded into navigation

space $\mathcal{N}$

$$\Psi : (\mathbf{R}, \mathbf{p}, t) \mapsto (\varphi, \theta, t).$$

In our system, Ψ is simply a transformation from Cartesian coordinates to spherical coordinates, where the sphere center $\mathbf{p}_S$ and the radius of the sphere r_S are computed from the extrinsic camera parameters $\mathbf{R}$ and $\mathbf{p}$ in a least-squares sense. The implications of this dimensionality reduction of $\mathcal{N}$ have been thoroughly discussed by Lipski et al. [19].

Partitioning the navigation space $\mathcal{N}$ into tetrahedra, each point $\mathbf{v}$ is defined by the vertices of the enclosing tetrahedron $\lambda = \{\mathbf{v}_i\}, i = 1 \dots 4$. Its position can be uniquely expressed as $\mathbf{v} = \sum_{i=1}^{4} \mu_i \mathbf{v}_i$, where μ_i are the barycentric coordinates of $\mathbf{v}$. Each of the 4 vertices $\mathbf{v}_i$ of a tetrahedron corresponds to a recorded image I_i. Each of the 12 edges e_{ij} correspond to a correspondence map $\mathbf{w}_{ij}$, that defines a translation of a pixel location $\mathbf{x}$ on the image plane.

3.2 Correspondence Fields

To be able to create image-based stereoscopic images it is important to get an insight into the nature of the information contained in the correspondence fields. A correspondence field is a dense vector field $\mathbf{w}_{ij}$ directed from recorded source image I_i to a destination image I_j. One option of creating those correspondence maps is the application of optical flow algorithms to the source and destination image.

In order to get a clear understanding of the information encoded within $\mathbf{w}_{ij}$, we assume a two camera setup. Figure 1 shows such a simplified setup where the cameras are restricted to a 1D movement along the horizontal axis. The vertical axis is the time, the dotted lines show the camera movement paths and each green dot corresponds to an image acquired at that specific time and place. In the following we investigate the information contained in the correspondence fields $\mathbf{w}_{12}$ and $\mathbf{w}_{13}$ when different constraints are posed on the recording modalities.

We start with the most restrictive camera setup with static cameras and synchronized camera shutters as shown in Fig. 1(a). The images are acquired on an axis aligned regular grid in the space-time plane. The correspondence field within the first camera $\mathbf{w}_{12}$ links two consecutive images of the video stream. Since the viewpoint does not change all image changes encoded within $\mathbf{w}_{12}$ represent motion of objects within the scene. In contrast the correspondence field $\mathbf{w}_{13}$ linking two adjacent cameras contains no object movement at all. Source and destination image have been acquired at the same point in time and therefore no object motion occurred. All changes along $\mathbf{w}_{13}$ can be explained by the motion parallax due to the changed viewpoint between cameras. If the source and destination image of $\mathbf{w}_{13}$ were rectified, the correspondence field would amount to a disparity map.

While this is not true in the general case, $\mathbf{w}_{13}$ strictly encodes the scene depth information and no object movement.

When the recording constraints are relaxed to a setup where the camera shutters are no longer synchronized, the interpretation of $\mathbf{w}_{13}$ changes. While the intra camera correspondence field $\mathbf{w}_{12}$ still encodes only object motion, the inter camera correspondence field $\mathbf{w}_{13}$ is no longer horizontally aligned with the time axis. In the space-time plane this amounts to a sheering of the two camera paths as depicted in Fig. 1(b). As a result of the unsynchronized shutters the objects in a dynamic scene can move between the acquisition times and the variation between source and destination image is no longer only defined by the view point change.

A similar change occurs when the cameras are allowed to change position. In the simplified one dimensional case the camera paths are no longer straight lines. As seen in Fig. 1(c) the direction from source to destination image for $\mathbf{w}_{12}$ is as well as $\mathbf{w}_{13}$ are no longer axis aligned. Both vector fields contain a mixture of scene depth and motion.

While we do not reconstruct an explicit depth or motion model of the scene at any point, the information is still encoded within the vector fields. This holds true for less simplified cases incorporating more cameras and view synthesis within $\mathcal{N}$.

Inherent to our model is however the restriction to linear motion. This applies on the one hand to the motion (Fig. 1(c)) of camera where $\mathbf{w}_{12}$ linearly connects source and destination images regardless of the actual camera motion. On the other hand it affects the object motion within the scene because a correspondence field defines a line along which each point can move on the image plane.

4 Stereoscopic Virtual View Synthesis

In this section, we recapitulate image-based virtual view synthesis and show how to use them to synthesize a stereoscopic image pair.

Following Lipski et al. [19], we synthesize the left view $I_v^L(\varphi, \theta, t)$ for every point $\mathbf{v}$ inside the recording hull of the navigation space $\mathcal{N}$ by multi-image interpolation:

$$I_v^L(\varphi, \theta, t) = \sum_{i=1}^{4} \mu_i \tilde{I}_i, \tag{1}$$

where

$$\tilde{I}_i \left(\Pi_i \mathbf{x} + \sum_{j=1,\ldots,4, j\neq i} \mu_j (\Pi_j (\mathbf{x} + \mathbf{w}_{ij}(\mathbf{x})) - \Pi_i \mathbf{x}) \right) = I_i(\mathbf{x}) \tag{2}$$

are the forward-warped images Mark et al. [22]. $\{\Pi_i\}$ defines a set of reprojection matrices $\{\Pi_i\}$ that map each image I_i onto the image plane of

$I_v^L(\varphi, \theta, t)$, as proposed by Seitz and Dyer [25]. Those matrices can be easily derived from camera calibration. Since the virtual image $I_v^L(\varphi, \theta, t)$ is always oriented towards the center of the scene, this re-projection corrects the skew of optical axes potentially introduced by our loose camera setup and also accounts for jittering introduced by dynamic cameras. Image re-projection is done on the GPU without image data resampling. As proposed by Stich et al. Stich et al. [26], disocclusions are detected on-the-fly by calculating local divergence in the correspondence fields. In contrast to their simple occlusion heuristic, we determine a geometrically valid disparity map. When applying the forward warp in a vertex shader program, we also determine where a given vertex would be warped to in the right eye's view. By subtracting both values, we derive a per-vertex disparity. This is interpolated in the fragment shader. Using this disparity value, occlusions can now be resolved by a simple depth buffer comparison. If necessary, the exact disparity of a parallel camera setup can be computed by applying the identical post-warping re-projection matrices. The view for the right eye $I_v^R(\varphi + \Delta, \theta, t)$ is synthesized similar to Eq. (1) by offsetting the camera position along the φ-direction. A common rule for stereoscopic capture is that the maximal angle between the stereo camera axes in a converging setup should not exceed 1.5 degrees Mendiburu [23]. Otherwise, the eyes are forced to diverge to bring distant objects in alignment which usually causes uncomfort. By nature of construction, our approach renders converging stereo pairs and angles of convergence between 0.5 and 1.5 degrees give the most pleasing stereoscopic results.

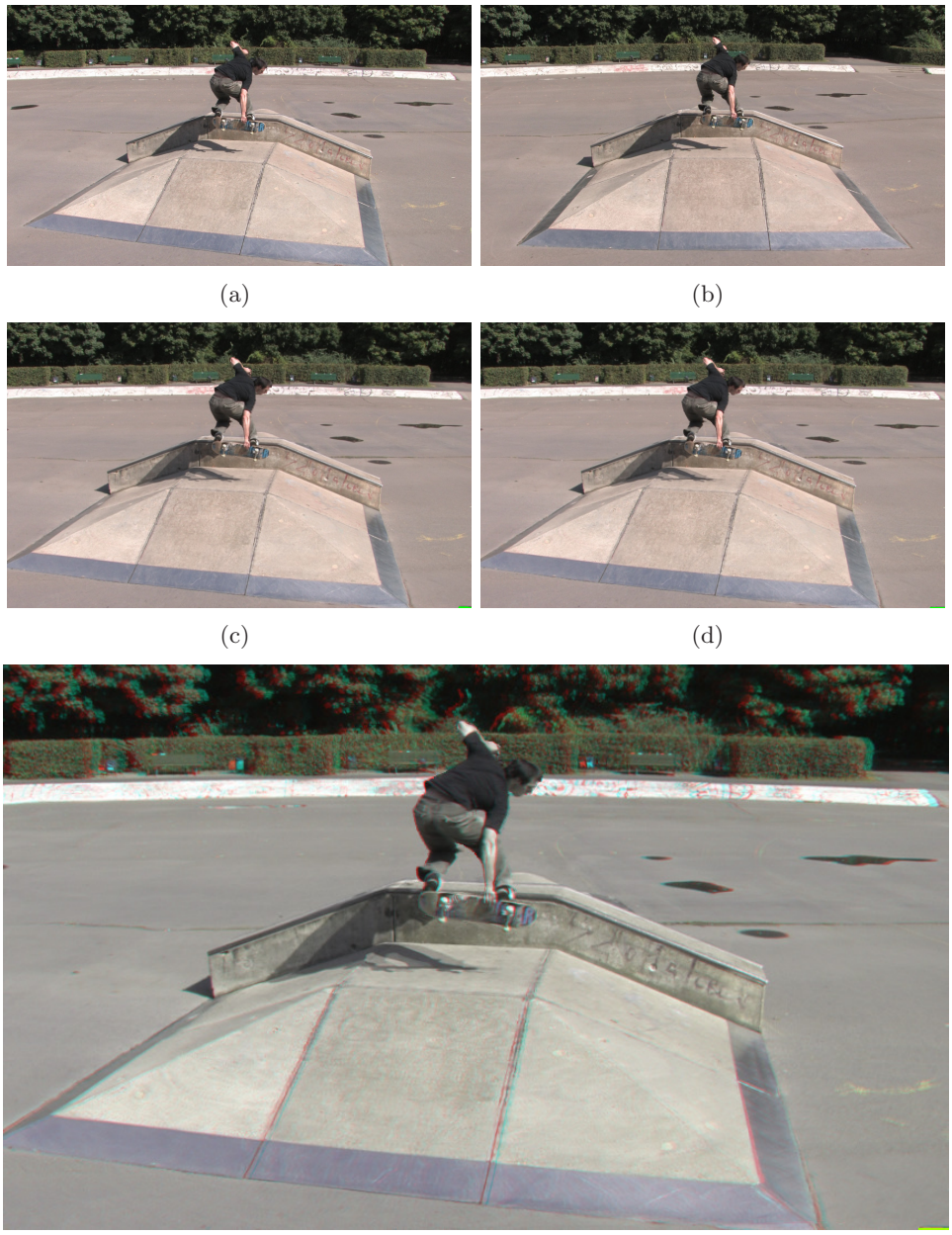
Our rendering algorithm is implemented on the GPU. At 940×560 pixels output resolution, rendering frame rates exceed 25 fps on an NVIDIA GeForce 295 GTX.

4.1 Disparity Effects

Our system is able to produce disparity-based visual effects, since disparity maps for each stereo pair are calculated on-the-fly for occlusion handling. While image-based systems often lack this crucial ability, a vast amount of editing tasks becomes possible with on-the-fly creation of disparity maps. Typical editing tasks are insertion of 3D objects with correct depth ordering, synthetic depth-of-field rendering, atmospheric effects (attenuation/fog) and refocussing on the object of interest. In Sect. 5 we show a selection of these effects. Of course, using this image-based disparity, any other non-linear mapping as presented in Lang et al. [15] can be applied as well.

4.2 Visual Effects

Space-time visual effects like the famous "Bullet Time" from the motion picture "Matrix" are created using the time-slice photography technique; a

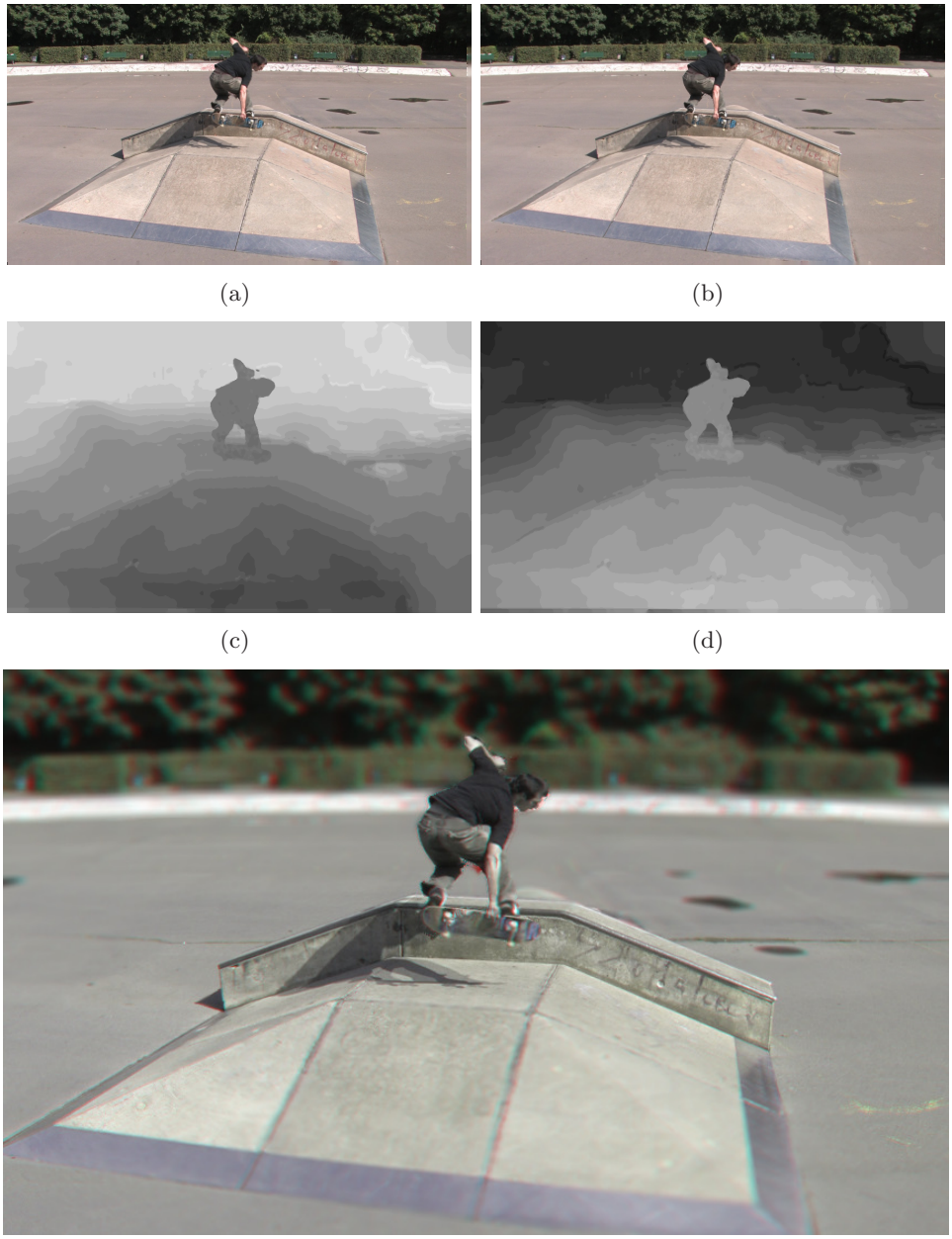


(a) (b)

(c) (d)

(e)

Figure 2: Multi-view Skater scene with one distinguished foreground object and large scene depth recorded with six cameras. (a),(b) Two original input views of adjacent cameras. (c),(d) Left and right view of stereo pair and (e) red-cyan anaglyph from virtual viewpoint.



(a) (b)

(c) (d)

(e)

Figure 3: Depth-of-field effect. (a),(b) show the left and right image of the synthesized stereo pair, (c),(d) visualize the disparity map from left to right and vice versa and (e) shows a red-cyan anaglyph with a depth-of-field effect based on the estimated disparities.

wide variety of other effects, e.g., *slow motion*, *motion blur*, *multi-exposure*, *motion distortion* and many more Digital Air Inc. [9], are created with a specific camera setup. Typically, each frame of the later effect sequence is recorded by a separate camera. On set, innumerous cameras must be exactly positioned and aligned, and shutter timings of all cameras must be precisely triggered. Obviously, an extension of those techniques to stereoscopic capture is even more tedious.

Linz et al. Linz et al. [18] recently showed how such effects can be created from a sparse unsynchronized camera setup. Being based on sparse multi-video footage, the embedding presented in Sect. 3.1, and making use of the same image synthesis framework as explained in Sect. 4, such effects can now also be created for stereoscopic devices with our proposed approach.

5 Results

All results in this section are presented in red (left) - cyan (right) anaglyph images. For further assessment of our results, we refer the reader to the accompanying video. In our experiments we used four different scenes to evaluate the performance of our algorithm. All setups were recorded with converging camera setups, i.e., the cameras were placed in an arc around the scene center.

The *Skater* scene, Fig. 2, is an outdoor recording featuring a high depth range. Figures 2(a) and (b) show the original input images for two adjacent cameras. The baseline in the setup is too wide and the image planes too different to make the original input views work as a stereo pair by themselves. In contrast, the final render results for the left and right stereo image in Figs. 2(c) and (d) are quite close. When watching the scene in motion, the stereo parallax effect of the actor in front of the background trees clearly enhances the already existing motion parallax, to give a very good impression of scene depth.

The *Parkour* scene, Fig. 4(a), exhibits great detail in the background of the scene. The twigs of the trees are sometimes only of sub-pixel size, are visible at multiple depths and occlude each other. Although our rendering model does not explicitly cope with this fine geometry, the rendered results are convincing. When viewing the stereoscopic output video, the parkour runner stands out clearly in front of the cluttered background, whereas he is more or less indistinguishable in a monocular sequence. It should also be apparent that it is very challenging to recover a 3D model of the scene. This applies equally to an automatic 3D reconstruction as well as to a manual modeling process. It can also be noted that the stereoscopic cues seem to convince the human eye of the plausibility of the scene. When watching a monocular rendering, rendering artifacts are spotted more easily.

In the *Graffiti* scene, Fig. 4(b), we show that we are not confined to

a static camera setup. Although the cameramen were moving during the recording, it is possible to render a static stereoscopic view of the scene.

For comparative purposes, we also rendered a stereoscopic sequence with the *Breakdancer* input material from Zitnick et al. Zitnick et al. [33], Fig. 4(c).

We demonstrate a combination of two disparity-based effects on the *Skater* scene. We refocus the stereoscopic image pair so that a virtual point of interest, i.e. a certain scene depth, has no disparity between the two eyes. A synthetic depth-of-field effect is further applied to steer the observer's attention towards these areas, Fig. 3(e). Stereoscopers may use similar effects to guide the observer's view and to strengthen or weaken the stereoscopic effect. We also integrated visual effects as described by Linz et al. Linz et al. [18] into our stereoscopic rendering. We rendered the Skateboard scene with a time-freeze (Fig. 2(e)), time-blur (Fig. 4(e)) and flashtrail effect (Fig 4(d)) to show the versatility of our approach.

6 Conclusion and Future Work

We presented an approach for stereoscopic free-viewpoint rendering that circumvents the need for explicit 3D reconstruction. It enables the flexible creation of stereoscopic content of complex natural scenes, where parameters as baseline, viewpoint and scene time can easily be modified in post production. In a single workflow, image alignment, free-viewpoint video and baseline editing can be performed.

Our approach can cope with asynchronously captured material and loosely calibrated camera setups greatly reducing the hardware requirements needed for stereoscopic 3D recording. A small subset of special effects were demonstrated and many possible effects that integrate seamlessly are conceivable. One future research direction could be to conceive an image-based editing framework, where the original footage could be manipulated with classic 2D tools.

References

[1] Baker, S., Roth, S., Scharstein, D., Black, M. J., Lewis, J., and Szeliski, R. 2007. A database and evaluation methodology for optical flow. In *IEEE International Conference on Computer Vision (ICCV'07)*. IEEE Computer Society, Los Alamitos, CA, USA, 1–8.

[2] Ballan, L., Brostow, G. J., Puwein, J., and Pollefeys, M. 2010. Unstructured video-based rendering: Interactive exploration of casually captured videos. *ACM Trans. on Graphics 29,* 3 (July), 87:1–87:11.

(a) Parkour scene.

(b) Graffiti scene.

(c) Breakdancer scene from Zitnick et al. [33].

(d) Skate scene with flashtrail effect.

(e) Skate scene with timeblur effect.

Figure 4: Rendered anaglyph stereo images. With our approach, stereoscopic images can be rendered from wide-baseline multi-view recordings. Please use red (left) - cyan (right) anaglyph glasses.

[3] Chen, S. E. and Williams, L. 1993. View interpolation for image synthesis. In *Proc. of ACM SIGGRAPH 1993*. ACM Press/ACM SIGGRAPH, New York, 279–288.

[4] de Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H.-P., and Thrun, S. 2008. Performance Capture from Sparse Multi-View Video. *ACM Trans. on Graphics 27,* 3, 98:1–98:10.

[5] Devernay, F. and Beardsley, P. 2010. *Image and Geometry Processing for 3-D Cinematography*. Vol. 5. Springer-Verlag, Chapter Stereoscopic cinema, 11–51.

[6] Devernay, F. and Duchêne, S. 2010. New view synthesis for stereo cinema by hybrid disparity remapping. In *Proc. of IEEE International Conference on Image Processing (ICIP'10)*. IEEE Computer Society, Hong Kong, 5–8.

[7] Devernay, F. and Peon, A. R. 2010. Novel view synthesis for stereoscopic cinema: Detecting and removing artifacts. In *ACM Workshop on 3D Video Processing (3DVP'10)*. ACM Press, Firenze, 25–30.

[8] Didyk, P., Ritschel, T., Eisemann, E., Myszkowski, K., and Seidel, H.-P. 2010. Adaptive image-space stereo view synthesis. In *Proc. of Vision, Modeling and Visualization (VMV 2010)*. Eurographics, Eurographics Association, Siegen, Germany, 299–306.

[9] Digital Air Inc. 2009. Digital Air Techniques. http://www.digitalair.com/techniques/index.html.

[10] Eisemann, M., Decker, B. D., Magnor, M., Bekaert, P., de Aguiar, E., Ahmed, N., Theobalt, C., and Sellent, A. 2008. Floating Textures. *Computer Graphics Forum 27,* 2 (4), 409–418.

[11] Germann, M., Hornung, A., Keiser, R., Ziegler, R., Würmlin, S., and Gross, M. 2010. Articulated billboards for video-based rendering. *Computer Graphics Forum 29,* 2, 585–594.

[12] Guillemaut, J.-Y., Kilner, J., and Hilton, A. 2009. Robust graph-cut scene segmentation and reconstruction for free-viewpoint video of complex dynamic scenes. In *IEEE International Conference on Computer Vision (ICCV'09)*. Kyoto, Japan, 809–816.

[13] Guillemaut, J.-Y., Sarim, M., and Hilton, A. 2010. Stereoscopic content production of complex dynamic scenes using a wide-baseline monoscopic camera set-up. In *Proc. of IEEE International Conference on Image Processing (ICIP'10)*. IEEE Computer Society, 9–12.

[14] Klose, F., Lipski, C., and Magnor, M. 2010. Reconstructing Shape and Motion from Asynchronous Cameras. In *Proc. of Vision, Modeling and Visualization (VMV 2010)*. Eurographics, Eurographics Association, Siegen, Germany, 171–177.

[15] Lang, M., Hornung, A., Wang, O., Poulakos, S., Smolic, A., and Gross, M. 2010. Nonlinear disparity mapping for stereoscopic 3d. *ACM Trans. on Graphics 29,* 3, 75:1–75:10.

[16] Lee, S., Wolberg, G., and Shin, S. 1998. Polymorph: morphing among multiple images. *IEEE Computer Graphics and Applications*, 58–71.

[17] Linz, C., Lipski, C., and Magnor, M. 2010. Multi-image Interpolation based on Graph-Cuts and Symmetric Optic Flow. In *Proc. of Vision, Modeling and Visualization (VMV 2010)*, C. R.-S. Reinhard Koch, Andreas Kolb, Ed. Eurographics, Eurographics Association, Siegen, Germany, 115–122.

[18] Linz, C., Lipski, C., Rogge, L., Theobalt, C., and Magnor, M. 2010. Space-Time visual effects as a Post-Production process. In *ACM Workshop on 3D Video Processing (3DVP'10)*. Firenze, Italy, 1–6.

[19] Lipski, C., Linz, C., Berger, K., Sellent, A., and Magnor, M. 2010. Virtual video camera: Image-based viewpoint navigation through space and time. *Computer Graphics Forum 29,* 8 (Dezember), 2555–2568.

[20] Lipski, C., Linz, C., Neumann, T., and Magnor, M. 2010. High Resolution Image Correspondences For Video Post-Production. In *Proc. of European Conference on Visual Media Production (CVMP'10)*. London, 33–39.

[21] Mahajan, D., Huang, F., Matusik, W., Ramamoorthi, R., and Belhumeur, P. 2009. Moving Gradients: A Path-Based Method for Plausible Image Interpolation. *ACM Trans. on Graphics 28,* 3, 42:1–42:11.

[22] Mark, W., McMillan, L., and Bishop, G. 1997. Post-Rendering 3D Warping. In *Proc. of Symposium on Interactive 3D Graphics (I3D'97)*. 7–16.

[23] Mendiburu, B. 2009. *3D Movie Making: Stereoscopic Digital Cinema from Script to Screen*. Focal Press.

[24] Rogmans, S., Lu, J., Bekaert, P., and Lafruit, G. 2009. Real-time stereo-based view synthesis algorithms: A unified framework and

evaluation on commodity gpus. *Signal Processing: Image Communication 24,* 1-2, 49 – 64. Special issue on advances in three-dimensional television and video.

[25] Seitz, S. M. and Dyer, C. R. 1996. View Morphing. In *Proc. of ACM SIGGRAPH 1996*. ACM Press/ACM SIGGRAPH, New York, 21–30.

[26] Stich, T., Linz, C., Albuquerque, G., and Magnor, M. 2008. View and Time Interpolation in Image Space. *Computer Graphics Forum 27,* 7, 1781–1787.

[27] Stich, T., Linz, C., Wallraven, C., Cunningham, D., and Magnor, M. 2010. Perception-motivated Interpolation of Image Sequences. *ACM Transactions on Applied Perception (TAP)*, to appear.

[28] Stich, T., Linz, C., Wallraven, C., Cunningham, D., and Magnorw, M. 2008. Perception-motivated Interpolation of Image Sequences. In *Proc. of ACM Symposium on Applied Perception in Graphics and Visualization (APGV)*. ACM, Los Angeles, USA.

[29] Vedula, S., Baker, S., and Kanade, T. 2005. Image Based Spatio-Temporal Modeling and View Interpolation of Dynamic Events. *ACM Trans. on Graphics 24,* 2, 240–261.

[30] Wilkes, L. 2009. The role of ocula in stereo post production. Tech. rep., The Foundry.

[31] Wolberg, G. 1998. Image morphing: a survey. *The Visual Computer 14,* 8, 360–372.

[32] Zilly, F., Müller, M., Eisert, P., and Kauff, P. 2010. The stereoscopic analyzer – an image-based assistance tool for stereo shooting and 3d production. In *Proc. of IEEE International Conference on Image Processing (ICIP'10)*. IEEE Computer Society, 4029–4032.

[33] Zitnick, C., Kang, S., Uyttendaele, M., Winder, S., and Szeliski, R. 2004. High-Quality Video View Interpolation Using a Layered Representation. *ACM Trans. on Graphics 23,* 3, 600–608.